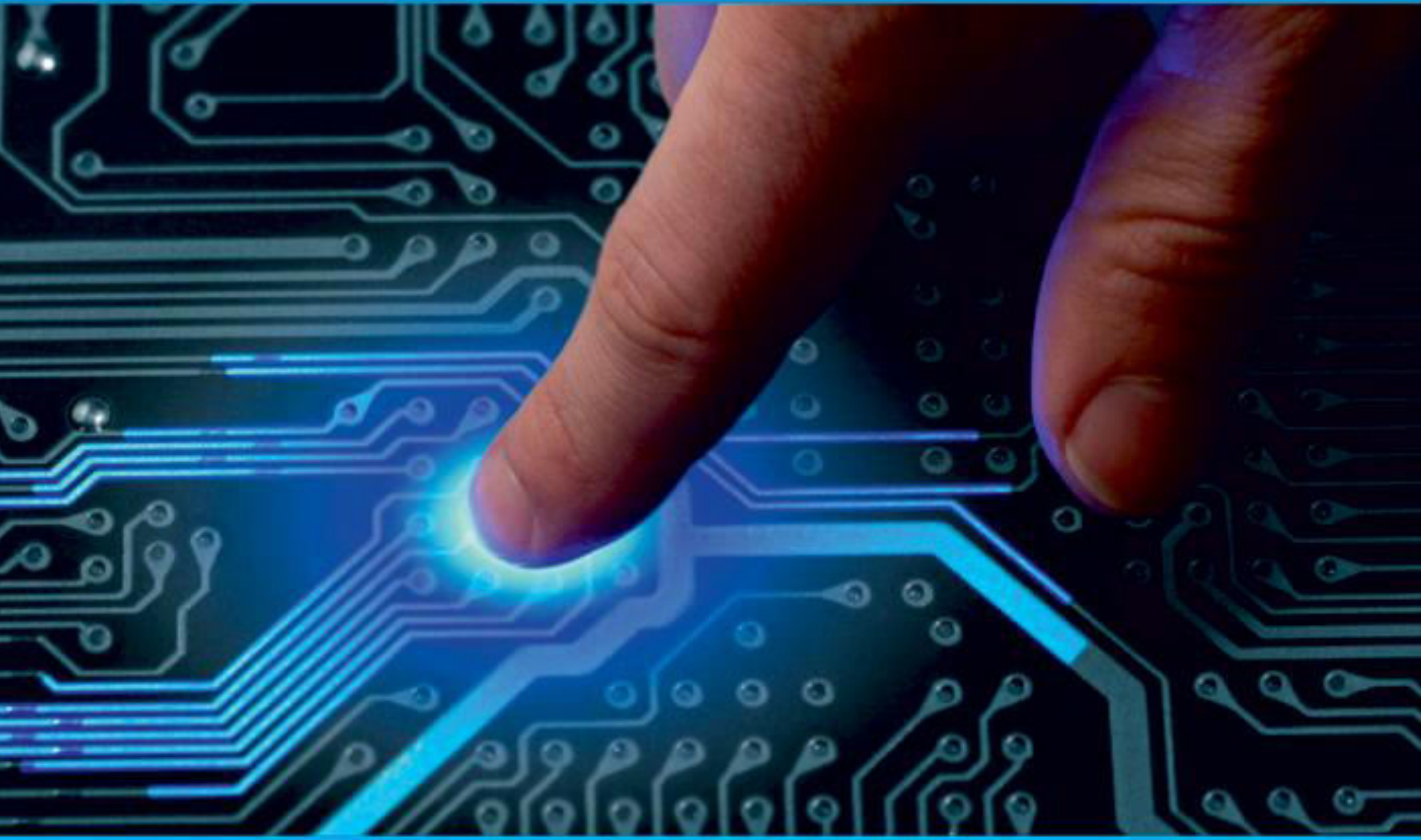
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 10, October 2023

INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.379

☎ 9940 572 462 6381 907 438 ijircce@gmail.com @ www.ijircce.com

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 10, October 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1110049 |

Federated Learning Trade-Offs: A Systematic Review of Privacy Protection and Performance Optimization

Prof. Neha Thakre[1], Prof. Nidhi Pateriya[2], Prof. Gulafsha Anjum[3], Divyanshi Tiwari[4], Aastha Mishra[5]

Department of Computer Science & Engineering, Badreia Global Institute of Engineering & Management, Jabalpur, Madhya Pradesh, India[1,2,3,4,5]

ABSTRACT: Federated Learning (FL) is an innovative approach in Artificial Intelligence (AI) that enhances privacy by avoiding centralized data storage and performing learning directly on users' devices. However, this method introduces new privacy concerns, especially during the training phase and when exchanging parameters between servers and clients. Although various privacy-preserving solutions have been developed to address these issues, integrating these mechanisms can lead to increased communication and computational overheads. This, in turn, may affect data utility and the performance metrics of learning systems. This paper presents a systematic literature review of key methods and metrics that help strike a balance between privacy and other performance aspects in FL applications, such as accuracy, loss, convergence time, utility, and overheads in communication and computation. The review offers a comprehensive overview of recent privacy-preserving techniques in FL across different applications, with a special emphasis on quantitative privacy assessment approaches. It aims to highlight the need for balancing privacy with practical requirements in real-world FL scenarios, while also identifying challenges, unresolved issues, and potential areas for future research.

KEYWORDS: Distributed artificial intelligence, Federated learning, Cybersecurity, Trustworthiness, Performance Evaluation.

I. INTRODUCTION

Centralized Machine Learning (ML) algorithms have revolutionized data management and analysis across various industries, enhancing operational efficiency, automating tasks, and providing deeper insights for better decision-making [46]. However, the extensive use of personal data by these centralized systems has raised privacy concerns, particularly since the implementation of the General Data Protection Regulation (GDPR) [142]. GDPR enforces strict regulations on the collection, storage, and processing of personal data, aiming to protect individuals' privacy rights and give them more control over their information. Federated Learning (FL) presents a promising approach to address these privacy concerns and comply with GDPR regulations [85, 139]. By avoiding data centralization and training ML models directly on user devices or edge servers, FL keeps sensitive data on the local device, thus ensuring privacy and security. Instead of sending raw data to a central server, FL transmits only local model updates, preserving the confidentiality of individual data points. This decentralized approach allows users to retain control over their personal information and decide whether to contribute to model training.

FL's privacy-preserving features have attracted significant interest across various sectors, including healthcare, finance, and the Internet of Things (IoT) [62]. In healthcare, FL enables medical institutions to train models with patient data while protecting sensitive information, thus supporting advancements in medical research, personalized medicine, and disease prediction while maintaining patient privacy [165, 18, 41]. In the financial sector, banks and financial institutions can use FL to develop predictive models without exposing customer financial data, such as transaction details and account balances, thus reducing the risk of data breaches [82]. FL is also widely used in IoT applications, allowing collaborative learning on distributed IoT devices and ensuring privacy in areas like smart homes, autonomous vehicles, and industrial sensors [79, 108, 117]. This enables model enhancement without compromising individuals' or organizations' privacy.

Despite its benefits, FL faces challenges related to the exchange of model update parameters between servers and clients [100]. This communication process could potentially allow adversaries to access and analyze model parameters, including the neural network's outputs, and reconstruct raw data through various attack methods. As the number of users and training iterations increases, the FL system becomes more vulnerable to a range of privacy attacks.

II. MOTIVATION

To our knowledge, there is currently no comprehensive survey that specifically focuses on methods and metrics for balancing privacy and performance in Federated Learning (FL). While there are existing surveys that address privacy-preserving techniques in FL, summarized in Table 1, they primarily concentrate on recent developments in privacy mechanisms and the associated risks, without evaluating how these mechanisms impact performance-related aspects. FL entails training local models on devices with limited resources, and implementing privacy-preserving measures on these devices can lead to increased costs and overhead, potentially compromising system performance in various ways.
The main aim of this paper is to review recent privacy-preserving mechanisms and assess their impact on performance-related application requirements. Additionally, this paper seeks to compile modern methods that propose strategies for achieving a balance between privacy and performance—an essential consideration in FL setups, as highlighted in the study [143]. This work also provides guidance for researchers on selecting suitable privacy-preserving mechanisms tailored to specific FL application domains.

Reference	Privacy-preserving categories	Privacy assessment	Balancing privacy-performance
	Encryption	Perturbation	Blockchain
X. Yin et al. [175]	✓	✓	✓
V. Mothukuri et al. [96]	✓	✓	
A. Blanco-Justicia et al. [14]	✓		✓
Our Work	✓	✓	✓

This table summarizes related surveys on privacy-preserving federated learning, highlighting the categories of privacy-preserving techniques they cover, whether they include privacy assessment, and if they balance privacy with performance.

III. MAIN CONTRIBUTIONS

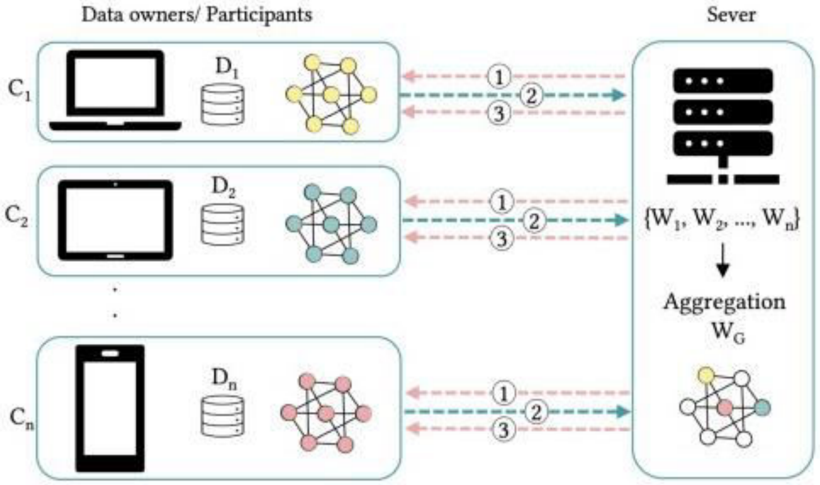
This paper provides a review of privacy-preserving mechanisms in federated learning (FL), emphasizing significant methods and metrics for balancing privacy with other performance-related requirements. We employ a comprehensive classification of recent publications, integrating existing categories from the literature and introducing previously overlooked or undervalued categories. This approach allows us to cover many new publications in the field. As outlined in Table 1, earlier surveys by V. Mothukuri et al. [96] and A. Blanco-Justicia et al. [14] primarily categorize FL privacy-preserving mechanisms into three distinct groups: encryption, perturbation, and blockchain. In contrast, X. Yin et al. [175] provides a broader overview of hybrid mechanisms but excludes blockchain from their categorization. This paper advances the categorization by presenting a structured analysis of privacy-preserving mechanisms within FL across four main categories: encryption, perturbation, blockchain, and hybrid. Furthermore, it breaks down these main categories into subcategories, offering a detailed exploration of the field that captures a wide array of recent scholarly contributions.

A thorough examination of the existing literature reveals the absence of a universally accepted metric or method to evaluate privacy in FL. Nonetheless, we have identified several metrics and assessed the effectiveness of privacy-preserving mechanisms against potential threats. These metrics can serve as benchmarks to measure a system's robustness against adversarial endeavors and provide a systematic method for evaluating prior work in this area. In the following sections, we provide an in-depth analysis of these metrics and methods. To the best of our knowledge, this paper is the first comprehensive analysis of privacy-preserving mechanisms in FL systems and their trade-offs with performance-related application requirements. We aim to shed light on how privacy considerations affect FL systems and their performance by addressing key research topics.
In summary, the main contributions of this Systematic Literature Review (SLR) are as follows:

- To determine the impact of privacy-preserving mechanisms in FL systems and their trade-offs with other performance-related application requirements based on a comprehensive review of existing literature.
- To investigate and evaluate existing methods and metrics for assessing the effectiveness of privacy-preserving mechanisms in FL, with a specific emphasis on quantitative assessment approaches.
- To categorize the latest research in FL privacy-preserving mechanisms, drawing from relevant scientific publications, and detail the various applications in which each mechanism has been applied.

IV. BASIC CONCEPTS OF FEDERATED LEARNING

Federated learning (FL), also known as collaborative learning, is a machine learning (ML) technique that trains an algorithm across multiple independent sessions, each using its own dataset. FL enables multiple participants, referred to as clients, to collaboratively develop a shared ML model without sharing their data. This approach addresses critical issues such as data privacy, data security, data access rights, and the use of heterogeneous data sources. FL was initially introduced by Google as a distributed training model executed on mobile devices, where local model updates are exchanged with a central server [85]. The server's primary role is to aggregate these local model updates to construct a global ML model. As illustrated in Fig. 2, the FL scenario assumes the presence of N clients, denoted as $C1$, $C2$, ..., Cn, each having access to their respective databases $D1$, $D2$, ..., Dn.



As shown in Fig. , FL training typically involves three steps:

- **Step 1**: The server broadcasts the initialized global model and assigns it to selected participants, specifying the hyperparameters of the global model and the training process, such as learning rate, batch size, and local epoch.
- **Step 2**: Participants use their local data and the initialized global parameters to update their local parameters. After minimizing each participant's loss function, the updated local model parameters are sent back to the server.
- **Step 3**: The server aggregates the local model parameters from each participant and sends the updated global model back to the participants.

Steps 2 and 3 are repeated until the global loss function converges or a desired training accuracy is achieved.

V. FEDERATED LEARNING SYSTEM MODEL AND DESIGN

This section aims to provide an overview of the core system model and design of federated learning (FL)

5.1 Data Partitioning

Federated learning (FL) can be classified into three types based on how data is distributed: horizontal FL, vertical FL, and transfer FL [171]. The type of data partitioning chosen depends on the use case, privacy needs, and dataset characteristics. The following sections elaborate on these categories:

5.2 Horizontal Federated Learning Horizontal FL involves participants that have different data samples but share the same feature space [179]. For example, in a scenario involving smartphones, each device may contain data from different users but have common features like user behavior and app usage. Google introduced a horizontal FL model for Android phones [86]. In this setup, a user updates the model parameters locally on their phone and uploads them to the Android cloud, allowing all data owners to contribute to a federated model based on similar feature dimensions

5.3 Vertical Federated Learning Vertical FL occurs when datasets contain the same samples or users but have different features [179]. Entity alignment is crucial in vertical FL to link these vertically partitioned datasets, enabling collaborative learning while maintaining data privacy and security. An example is healthcare data, where one device may have patient demographics and another device may have medical histories. By combining models trained on each device, a global model can be built without exposing raw data. Vertical FL is also used in finance, such as collaborations between Webank and invoice agencies to develop financial risk models with overlapping users but distinct features [20].

5.4 Federated Transfer Learning When there is limited overlap between users and features in two datasets, federated transfer learning is used to bridge the gap instead of dividing the data. This approach transfers knowledge from a party with a rich feature space to another party lacking sufficient features or labels to train a high-performing model [78]. For instance, radiology departments may struggle to gather enough scans to develop an accurate diagnostic system. Using transfer learning, radiologists can improve diagnoses by leveraging insights from related tasks like image recognition [179].

VI. CLIENT SELECTION

The selection of clients in federated learning (FL) depends on various factors such as charging status and network connectivity. A common method to initiate communication and register participation is using a random number to select among these devices. However, this method has drawbacks, especially with a diverse range of clients, leading to longer training times. Various research efforts have proposed solutions to these challenges [3].

One proposed solution is a novel FL protocol called FedCS, which aims to actively manage resources for heterogeneous clients within mobile edge computing frameworks [104]. FedCS introduces specific deadlines for clients to download, update, and upload ML models. By aggregating updates from as many clients as possible within a limited time frame, the ML training process becomes more efficient. This approach significantly reduces training time and enhances the overall efficiency of the FL process. Additionally, FedCS considers factors such as the limited computing resources on client devices, ensuring that the training process adapts to the capabilities of individual clients.

However, when applied to Speech Emotion Recognition (SER) applications, Local Differential Privacy (LDP) does not provide acceptable accuracy due to the negative impact of adding noise to voice data, which can distort the audio signal [14]. Additionally, introducing noise to SER model parameters can compromise the model's utility by distorting or misaligning the parameters, leading to errors in the model's output [15]. This loss in accuracy is particularly harmful for most SER applications, which require precise results for industrial use [16]. Therefore, developing LDP mechanisms in Federated Learning (FL) for SER necessitates finding a solution that effectively reduces the impact of noise on SER accuracy while maintaining strong privacy protections.

This paper proposes a method called LDP-FL with Client Selection Strategy (CSS), which combines LDP with a novel client selection strategy to enhance privacy while preserving acceptable accuracy in SER within the FL system. LDP is used to protect clients’ speech datasets, while CSS minimizes the negative impact of noise scaling on model updates, resulting in more representative updates and improved accuracy.

Moreover, our study focuses on adapting the model inversion attack, initially developed for facial recognition models [17], for the SER model by making appropriate configuration adjustments. This attack attempts to reconstruct speech features by leveraging an adversary's knowledge of a specific client's emotion label and their local SER model. The main objective is to evaluate the effectiveness of the LDP method in protecting against such attacks within the FL setup.

Finally, we comprehensively evaluated the LDP-FL with CSS approach, focusing on its alignment with SER requirements and analyzing the trade-off between accuracy and privacy.
The novel contributions of this paper can be summarized as follows:

- We introduce a novel approach that integrates local differential privacy in federated learning (LDP-FL) with a client selection strategy (CSS) to enhance privacy while reducing the impact of noise on SER accuracy.
- We implement model inversion attacks to assess the robustness of LDP-FL and evaluate its effectiveness in preserving privacy. These attacks involve an adversary's attempt to reconstruct individuals' voice samples based on the output labels provided by the SER model.
- We conduct a comprehensive evaluation of the LDP-FL with CSS approach on public SER datasets, considering critical parameters such as privacy budget, noise scale, failure probability, and clipping threshold value. Our evaluation focuses on assessing how well our method meets SER requirements and analyzing the balance between accuracy and privacy.

VII. BACKGROUND AND RELATED WORKS

In this section, we provide an overview of the background and related work on Local Differential Privacy (LDP) mechanisms in Federated Learning (FL). We will also discuss the use of FL for Speech Emotion Recognition (SER) applications and related research.

7.1 Privacy-preserving Federated Learning

Federated Learning (FL) protects user privacy by decentralizing data storage from a central server to edge devices. However, sharing information such as model weights with servers can still pose privacy threats [8]. In FL, the exchange of model update parameters between central servers and clients can be exploited by attackers. In a white-box scenario, attackers can access the model, its architecture, weight parameters, and any necessary hyperparameters for predictions. In a black-box scenario, the adversary can only observe the model's outputs on arbitrary inputs [9].
Local Differential Privacy (LDP) has become a popular technique for preserving privacy in FL [10]. LDP can prevent individual device data from being leaked to the central server during the model training process [11], [12]. This technique involves adding artificial noise to each model's updated parameters before sharing them with the central server. Recent work proposed a framework called NbAFL, which utilized LDP and demonstrated its ability to meet differential privacy requirements under different protection levels by appropriately adjusting the variances of artificial noise [12]. Another study introduced LDP-based stochastic gradient descent (SGD) that ensures a given LDP level by setting a noise variance limit after multiple rounds of weight updates, using a tight composition theorem [13].

7.2 Speech Emotion Recognition using Federated Learning

Speech Emotion Recognition (SER) technology aims to identify and understand human emotions through speech. SER systems analyze audio signals from human speech and utilize machine learning algorithms to detect patterns and classify the emotional states conveyed in the speech [2]. Building SER models requires substantial amounts of data, which often includes sensitive personal information such as speech signals and emotions. However, storing this data centrally poses privacy risks. To mitigate these risks, federated learning (FL) offers a promising solution by allowing models to be trained collaboratively on decentralized devices without transferring raw data [7].
A paper [18] introduces an FL-based approach for developing a private decentralized SER model. This method uses data-efficient federated self-training to train SER models with minimal on-device labeled samples. However, the method relies solely on the FL framework as a privacy-preserving technique and does not consider threat models from clients or servers in FL, nor does it incorporate other privacy-preserving techniques. Similarly, another study [19] proposes a federated adversarial learning framework to protect both data and deep neural networks in SER. This framework includes an FL component for data privacy and adversarial training to enhance model robustness during the training phase. However, like the previous method, it only relies on the FL framework for privacy preservation and does not take into account other privacy-preserving techniques in FL.

VIII. SYSTEM DESCRIPTION

This section will cover the non-functional requirements of the SER (Speech Emotion Recognition) application (III-A), outline the relevant threat model (III-B), introduce the proposed LDP-FL with CSS method (III-C) along with its algorithms and details, and finally, examine the model inversion attack on speech features using specific algorithms (III-D).

8.1 Non-Functional Requirements of the Speech Emotion Recognition Application

Non-functional requirements pertain to the overall characteristics or quality attributes of a system that affect its performance rather than its specific functions. For SER applications, key non-functional requirements include privacy and accuracy. Meeting these requirements is crucial to address user needs and expectations while adhering to legal standards. This section will provide a detailed explanation and describe how these requirements are addressed in the evaluation section (IV).

1. **Privacy**
   a) Personal speech data must remain solely on local devices [5].
   b) The central server or any potential eavesdropper should not be able to extract sensitive information from the local model parameters.
2. **Accuracy**
   a) The accuracy of SER (Speech Emotion Recognition) applications should be sufficiently high to reliably identify emotions from speech samples. A baseline accuracy of at least 70% is considered acceptable for detecting the four primary emotions: neutral, sad, happy, and angry [20].

It's important to note that these requirements can be highly interconnected. For example, privacy-preserving measures can affect accuracy due to factors like distributed setups in federated learning (FL) or the noise introduced by the Local Differential Privacy (LDP) method. Furthermore, implementing SER in an FL environment may lead to an accuracy decrease of 0-5% as noted in reference [18].

8.2 Threat Model

This paper operates under the assumption that the server adheres to the honest-but-curious (HBC) model. In this scenario, the server complies with the federated learning (FL) protocol and does not engage in malicious activities, but it may still be interested in the data or models of other clients. Although client data remains stored locally in FL, intermediate parameters, such as wiw_iwi, must be shared with the server. This exchange has the potential to reveal private information about clients, as demonstrated by model inversion attacks. For example, a study [17] illustrated how a model inversion attack could reconstruct images from a facial recognition system.

8.3 Proposed Method: LDP-FL with CSS

We introduce a new approach called "LDP-FL With CSS," which integrates local differential privacy (LDP) with a client selection strategy (CSS) within federated learning to optimize both client privacy and the accuracy of the SER model. Our method addresses privacy requirement 1.a by ensuring that speech data is processed and trained locally on clients' devices within the federated learning framework. By using LDP techniques, Gaussian noise is added to the local updates before they are sent to the central server. This approach ensures strong protection against the inference of sensitive information, meeting requirement 1.b and reducing potential risks in the threat model.

To address accuracy requirement 2.a, we use CSS to prioritize clients with larger datasets and include them in each training round. This helps to counteract the possible negative effects of LDP on accuracy, aiming to improve it to the desired levels.

Figure 1 illustrates our proposed method, which involves three key steps. First, the server sends out the initial global SER model and applies the CSS to select clients for training. In the second step, the selected clients process speech data, extract Emobase features (as detailed in Sec. IV-A), and train their local models. They update their local model parameters with the global model and apply the LDP technique to these parameters before sending them to the server. In the final step, the server aggregates the noisy local model parameters and updates the global model, which is then sent back to the clients. A detailed description of the LDP-FL with CSS approach is provided in Algorithm 1. We will now explore the concepts of LDP and CSS within the context of federated learning.

IX. ALGORITHM 1: LDP-FL WITH CSS

```
Input: Number of iterations: T, Number of selected
clients: K, Local minibatch size: B, Initial
global model: w0, Learning rate: η, Clipping
threshold: C, LDP parameters: ϵ and δ
1 Initialization:
2 Initialize the global model parameters w0
for t ≤ T do
3 The server broadcasts current model wt
4 K: Client Selection Strategy (CSS)
5 Clients-side:
for i ∈ 1, 2, ...,K do
6 for each batch b ∈ Bi do
7 Compute gradient g(b) ←− ∇wLi(wt; b)
8 Clip gradient g(b) ←− g(b)/Max(1, ∥g(b)∥
C
)
9 Add Noise
˜ gi =
1
|B|
(Pb∈B g(b) + N(0, σ2C2I)
10 Share ˜ gi with server
11 Server-side:
12 Aggregate ˜g = 1
K PK
i=1 ˜ gi
13 Global model update wt+1 ←− wt − η.˜
```

X. LOCAL DIFFERENTIAL PRIVACY (LDP)

Local Differential Privacy (LDP) is a privacy-preserving framework where users do not need to trust any external party, including the central data collector. In this setup, users themselves apply random perturbations to their data to ensure privacy. Each user processes their data using a random perturbation algorithm, denoted as MMM, and shares the perturbed data with an aggregator or central server. The privacy budget ϵ\epsilonϵ defines the level of privacy protection, with a higher ϵ\epsilonϵ indicating lower privacy. The parameter δ\deltaδ represents the likelihood that the LDP mechanism fails to meet the privacy guarantee.

Formally, LDP is defined as follows:

10.1 Definition 1 ((ϵ,δ\epsilon, \deltaϵ,δ)-LDP [22]): A randomized mechanism MMM satisfies (ϵ,δ\epsilon, \deltaϵ,δ)-LDP if for any input values vvv and v′v'v′ in the domain of MMM, and for any output y∈Sy \in Sy∈S, the following holds:
Pr[M(v)=y]≤eϵ⋅Pr[M(v′)=y]+δ.\text{Pr}[M(v) = y] \leq e^{\epsilon} \cdot \text{Pr}[M(v') = y] + delta.Pr[M(v)=y]≤eϵ⋅Pr[M(v′)=y]+δ.
In theoretical terms, (ϵ,δ\epsilon, \deltaϵ,δ)-LDP means that the mechanism MMM provides the privacy guarantee with probability at least 1−δ1 - \delta1−δ.
To apply the LDP mechanism in a federated learning (FL) context, we followed the method outlined in reference [23]. Specifically, we added Gaussian noise to the clients' model parameters. To ensure the noise distribution Z∼N(0,σ2C2I)Z \sim N(0, \sigma^2C^2I)Z∼N(0,σ2C2I) adheres to (ϵ,δ\epsilon, \deltaϵ,δ)-LDP, we choose the noise scale σ≥cqTlog(1/δ)ϵ\sigma \geq c \frac{q \sqrt{T \log(1/\delta)}}{\epsilon}σ≥cϵqTlog(1/δ), where ccc is a constant and qqq is the sampling probability, with ϵ<cq2T\epsilon < cq^2Tϵ<cq2T and δ>0\delta > 0δ>0. Here, ZZZ represents the additive noise for the client gradient.
In Algorithm 1, during time slot ttt, each selected client i∈ki \in ki∈k trains its local dataset by minimizing the loss function ∇Li\nabla L_i∇Li (lines 6-8). The gradient g(b)g(b)g(b) for each b∈Bib \in B_ib∈Bi is calculated. To control the effect of large gradients, we use gradient clipping with the ∥L∥2\|L\|_2∥L∥2 norm. Specifically, the gradient

g(b)g(b)g(b) is replaced with g(b)/max(1,‖g(b)‖2/C)g(b)/\max(1, \|g(b)\|_2 / C)g(b)/max(1,‖g(b)‖2/C), where CCC is the clipping threshold (line 7). This ensures that if ‖g‖2\|g\|_2‖g‖2 is below CCC, the gradient remains unchanged. If ‖g‖2\|g\|_2‖g‖2 exceeds CCC, it is scaled down to maintain a norm of CCC, thus controlling the influence of large gradients.

After clipping, we compute the average of all gradients in set BBB and add scaled Gaussian noise Z∼N(0,σ2C2I)Z \sim N(0, \sigma^2C^2I)Z∼N(0,σ2C2I) to each client's gradient to achieve LDP (lines 9-10). The resulting noisy gradient g~i\tilde{g}_ig~i is then sent to the server (line 11). On the server side, the noisy gradients g~i\tilde{g}_ig~i from the selected clients are aggregated using the FedSGD algorithm, resulting in g~=1K∑i=1Kg~i\tilde{g} = \frac{1}{K} \sum_{i=1}^{K} \tilde{g}_ig~=K1∑i=1Kg~i. The global model is then updated with Wt+1←Wt−η·g~W_{t+1} \leftarrow W_t - \eta \cdot \tilde{g}Wt+1←Wt−η·g~ and used for the next iteration (lines 13-14).

10.2 Client Selection Strategy (CSS)

To address potential noise issues and maintain the accuracy of SER models in federated learning (FL), we propose a refined client selection strategy called Client Selection Strategy (CSS). This strategy involves a two-step selection process for clients participating in FL training.

XI. ALGORITHM 2: CLIENT SELECTION STRATEGY (CSS)

Input: Number of iterations: TTT, List of clients: LLL, Number of selected clients: KKK

Output: List of selected clients

1. For ttt ≤ TTT:
2. Select half of the clients: M=K/2M = K/2M=K/2
3. Sort LLL in descending order by sample size to get CCC
4. Choose the top MMM clients from CCC
5. Randomly select the remaining clients from C[M:]C[M:]C[M:]
6. Return the combination of selected and randomly chosen clients

Initially, we select half of the clients based on their dataset size, prioritizing those with larger sample sizes. This approach ensures that clients with more extensive local datasets, which are likely to provide more accurate and representative model updates, are favored. The remaining clients are chosen randomly to introduce diversity and minimize potential selection bias from relying solely on sample size.

Algorithm 2 details how the CSS method is applied in each training round of the overall Algorithm 1. The strategy first selects the top half of clients based on dataset size (line 4), giving preference to those with larger datasets. To prevent bias, the remaining clients are chosen randomly (line 5). The final list of selected clients combines both sets, ensuring no overlap and that each client is selected exactly once per training round (line 6).

By using the CSS approach, we balance the advantages of large local datasets with the need for diversity in client selection. This method helps minimize the impact of noise and preserves the initial accuracy of SER models trained through FL.

XII. MODEL INVERSION ATTACK FOR SPEECH EMOTION RECOGNITION MODELS

A model inversion attack occurs when an attacker gains access to a model's outputs or parameters with the aim of deducing sensitive training data. In our study, we adapt a model inversion attack methodology initially developed for face recognition [17] to the domain of speech emotion recognition by modifying certain parameters. In this scenario, we assume the attacker knows a specific emotion label—such as neutral, sad, happy, or angry—and has access to the model used by the clients. The attacker's goal is to reconstruct the speech features associated with a particular client and their corresponding emotion label.

The focus of the model inversion attack here is on reconstructing speech features, which encapsulate the high-level statistical characteristics of a client's speech. Each feature intensity is represented by a floating-point value. We assume the attacker does not have precise knowledge of the feature values they are attempting to infer. The feature vectors are modeled with nnn components and four possible emotion labels. Each emotion recognition classifier is represented as a function f~:[0,1]n→[0,1]4\tilde{f} : [0, 1]^n \to [0, 1]^4f~:[0,1]n→[0,1]4, where the output is a probability vector indicating the likelihood of each feature vector belonging to a specific emotion label. The notation f~label(x)\tilde{f}_{\text{label}}(x)f~label(x) refers to the iii-th component of this output vector.

XIII. ALGORITHM 3: MODEL INVERSION ATTACK FOR SPEECH EMOTION RECOGNITION MODELS

Input: Number of iterations: TTT, Best score: γ\gammaγ, Target model: f~\tilde{f}f~, Learning rate: η\etaη

Output: Reconstructed speech features for the target label

1. Set c=1−f~label(x)c = 1 - \tilde{f}_{\text{label}}(x)c=1−f~label(x)
2. Initialize x0←0x_0 \leftarrow 0x0←0
3. For ttt ≤ TTT: 4. Update xt←Process(xt−1−η·∇c(xt−1))x_t \leftarrow \text{Process}(x_{t-1} - \eta \cdot \nabla c(x_{t-1}))xt←Process(xt−1−η·∇c(xt−1)) 5. If c(xt)≥max(c(xt−1),…,c(xt−β))c(x_t) \geq \max(c(x_{t-1}), \ldots, c(x_{t-\beta}))c(xt)≥max(c(xt−1),…,c(xt−β)), then break 6. If c(xt)≤γc(x_t) \leq \gammac(xt)≤γ, then break
4. Return [argminxt(c(xt)),minxt(c(xi))][\text{argmin}_{x_t}(c(x_t)), \text{min}_{x_t}(c(x_i))][argminxt(c(xt)),minxt(c(xi))]

This algorithm uses gradient descent to minimize a cost function related to the emotion recognition model f~\tilde{f}f~. The process iteratively updates the candidate solution by moving in the direction opposite to the gradient. The cost function ccc is defined based on f~\tilde{f}f~.

The attack employs gradient descent with a maximum of TTT iterations and a step size of η\etaη. After each iteration, the resulting feature vector undergoes a post-processing step via a function called "Process," which may involve adjustments like denoising or sharpening. The descent process terminates if no improvement in cost is observed within β\betaβ iterations or if the cost surpasses the threshold γ\gammaγ. The best candidate solution is then returned.

XIV. EXPERIMENT RESULTS

In this section, we outline an industrial use case and the simulation setup for evaluating the performance of the LDP-FL method on Speech Emotion Recognition (SER) accuracy. We examine how factors such as noise scale, failure probability, and clipping threshold impact SER accuracy. Additionally, we assess the effect of Client Selection Strategy (CSS) within the LDP-FL framework and evaluate the robustness of LDP-FL against model inversion attacks. Finally, we explore the critical balance between privacy and accuracy.

A. USE CASE DESCRIPTION AND SIMULATION SETUP

The DAIS1 (Distributed Artificial Intelligent System) [24] project is a pan-European initiative focused on ensuring reliable connectivity and interoperability by integrating IoT with AI in a distributed edge system for industrial applications. The project encompasses use cases in digital life, digital industry, and smart mobility. A key use case within DAIS1 involves SER for TV recommendation systems, where the goal is to accurately detect user emotions and provide tailored movie recommendations to enhance user satisfaction. This requirement drives the exploration of LDP-FL with CSS for SER.

In our study, we evaluated the proposed method using the CREMA-D dataset [25], a widely utilized SER dataset comprising 7,442 clips from 91 actors. The dataset features clips from 48 male and 43 female actors aged between 20 and 74, representing various racial and ethnic backgrounds (African American, Asian, Caucasian, Hispanic, and Unspecified). Each actor recorded 12 different sentences, each expressed with one of six emotions (Anger, Disgust, Fear, Happy, Neutral, and Sad) and four emotion levels (Low, Medium, High, and Unspecified). For training the SER model, we focused on the four most prevalent emotion labels: neutral, sad, happy, and angry.

We used the OpenSMILE toolkit [26] to extract the Emo-Base feature set, which is commonly employed for SER tasks. These features capture a range of acoustic characteristics associated with different emotions and are known for their effectiveness in emotion recognition. Following feature extraction, we employed a multilayer perceptron (MLP) model for SER training using the FedSGD algorithm. The MLP architecture comprises two dense layers with sizes [256, 128] and uses ReLU activation functions, along with a dropout rate of 0.2. We set the local training batch size to 20 and the learning rate to 0.1 to facilitate faster convergence in FedSGD.

In the FL training process on the CREMA-D dataset, each speaker acts as an individual client due to the dataset's 91 distinct speakers. We allocated 80% of the data for local training and reserved 20% for validation. To ensure robustness, we conducted five experiments with different test folds and reported the average results from these five-fold experiments. The FL training was performed over 200 global epochs. All experiments were executed in a Windows 10 Pro environment with an Intel® Core™ i7 CPU @1.80GHz and 16.0 GB of RAM.

B. SER ACCURACY RESULTS ACROSS DIFFERENT PARAMETERS: NOISE SCALE, FAILURE PROBABILITY, AND CLIPPING THRESHOLD

We analyzed the impact of various parameters on the accuracy of SER within the LDP-FL framework. This evaluation involved 50 randomly selected clients and 120 training epochs, as shown in Figure 2. We specifically assessed how different noise scales (σ\sigmaσ), failure probabilities (δ\deltaδ), and clipping thresholds (CCC) affect accuracy.

Figure 2(a) illustrates that as the number of training epochs increases, the accuracy of LDP-FL stabilizes, indicating that the method converges. However, a higher noise scale, such as σ=10\sigma = 10σ=10, can disrupt convergence due to the excessive noise introduced during training, resulting in system instability. Figure 2(b) shows that higher failure probabilities (δ\deltaδ) lead to quicker convergence and improved accuracy, though at the cost of reduced privacy protection. In contrast, lower failure probabilities offer better privacy guarantees but lower accuracy. For example, a failure probability of δ=10−3\delta = 10^{-3}δ=10−3 achieved the highest accuracy, though it compromised some level of privacy.

Our assessment of LDP-FL with various clipping thresholds indicated that thresholds of 1.0 or 2.0 provided high accuracy and fast convergence, as depicted in Figure 2(c). However, increasing the threshold beyond a certain point resulted in decreased accuracy due to excessive information loss during the clipping process. Therefore, selecting an optimal clipping threshold is essential for balancing privacy and model accuracy.

C. EFFECT OF CSS ON SER ACCURACY

To evaluate the impact of the Client Selection Strategy (CSS) on SER performance, we compared CSS with the commonly used random selection (RS) method in both LDP and non-LDP federated learning systems. Using parameters σ=1.0\sigma = 1.0σ=1.0, C=2C = 2C=2, δ=10−5\delta = 10^{-5}δ=10−5, and K=50K = 50K=50, we observed a notable increase in accuracy from 60% to 70% when employing CSS with LDP, as illustrated in Figure 3. This improvement meets the accuracy requirements outlined in Section III-A. CSS proved effective in selecting clients, leading to larger and more representative datasets for training, which enhanced model robustness and accuracy.
However, it is important to note that selecting clients with larger local datasets can increase their exposure, potentially leading to data leakage. Hence, a careful balance is needed when using CSS.

Interestingly, the choice of client selection method did not significantly affect the accuracy of non-LDP FL systems. This suggests that the accuracy benefits of CSS are particularly relevant to LDP-FL scenarios. Therefore, adopting a client selection strategy like CSS can significantly improve LDP-FL performance and help mitigate the negative impact of LDP on accuracy.

D. ROBUSTNESS OF LDP-FL AGAINST MODEL INVERSION ATTACKS

We assessed the robustness of the LDP-FL method against model inversion attacks using the defined attack settings: T = 200, η = 0.1, β = 100, and γ = 0.99. The attack was conducted under two scenarios: LDP-FL with clipping thresholds C=[1,2,4]C = [1, 2, 4]C=[1,2,4] and failure probability δ = 10^{-5}, and a non-LDP FL setup with K=7K = 7K=7. We performed the attack on various client models and labels, and the average results were reported.

The goal of the model inversion attack is to reconstruct client speech features by exploiting the local SER model and its associated labels. To measure the attack's effectiveness, we used the Mean Squared Error (MSE) metric, comparing the reconstructed speech features with the actual features of each client.

The results, shown in Table I, indicate that at a noise scale σ\sigmaσ of 1.0, the MSE values were similar for both LDP and non-LDP setups. However, as we increased σ\sigmaσ and the clipping threshold CCC, the MSE values rose significantly, signaling a decrease in attack effectiveness. This demonstrates that incorporating LDP enhances privacy by effectively countering model inversion attacks. LDP significantly reduces the adversary's ability to accurately reconstruct speech features, thereby aligning with the privacy requirements, particularly 1.b.

E. BALANCING PRIVACY AND ACCURACY

Finding the optimal balance between privacy and accuracy is crucial when applying LDP to SER applications, which demand precise results. According to reference [23], the parameter epsilon (ϵ) quantifies the level of privacy protection offered by the (ϵ, δ)-LDP mechanism, with lower ϵ values indicating stronger privacy.
In our method, ϵ is influenced by the number of training epochs (T). As the number of epochs increases, ϵ adjusts, even if the noise scale remains constant. This relationship is depicted in Figure 4.
For the LDP-FL with CSS mechanism in SER, we experimented with various noise scales (σ\sigmaσ), k=50k = 50k=50, failure probability $\delta = 10^{-5}$, and a clipping threshold C=2C = 2C=2 over 50 epochs (T). The findings, illustrated in Figures 4 and 5, are as follows:

- With σ=5\sigma = 5σ=5, the privacy level achieved was (1.08, 10^{-5})-LDP, with an accuracy of approximately 54%.
- With σ=4\sigma = 4σ=4, the privacy level was (1.39, 10^{-5})-LDP, and accuracy improved to about 64%.
- Using σ=3\sigma = 3σ=3, we reached a privacy level of (1.92, 10^{-5})-LDP and an accuracy of around 67%.
- A privacy level of (3.51, 10^{-5})-LDP was obtained with σ=2\sigma = 2σ=2, resulting in an accuracy of approximately 69%.
- Finally, with σ=1\sigma = 1σ=1, the privacy level was (9.69, 10^{-5})-LDP, and accuracy was roughly 70%.

Achieving the right balance between privacy and accuracy depends on specific system requirements. For the SER application in the FL setup, where the acceptable accuracy range is 65-70% and privacy requirements are detailed in Section III-A, it is possible to achieve an acceptable privacy level with a parameter of (1.92, 10^{-5})-LDP and a noise scale of σ=3\sigma = 3σ=3, while maintaining the desired accuracy.

XV. CONCLUSIONS

In this study, we presented a new method, LDP-FL with CSS, designed to enhance privacy for SER applications while preserving system accuracy. This approach integrates Local Differential Privacy (LDP) with a Client Selection Strategy (CSS) to address the challenge of maintaining accuracy despite the introduction of privacy-preserving noise. By strategically selecting clients based on their dataset size for each Federated Learning (FL) training round, we were able to mitigate the negative impact of noise on accuracy.

Our evaluation using the CREMA-D dataset showed that LDP-FL with CSS achieves an accuracy between 65-70%. This is slightly lower than the initial accuracy of the SER model but meets the privacy standard of (1.92, 10^{-5})-LDP. The results underscore the importance of balancing privacy and accuracy to meet the specific needs of SER applications.

Looking ahead, we aim to explore personalized privacy solutions by adapting the noise scale of LDP mechanisms to align with individual client privacy preferences.

ACKNOWLEDGEMENTS

This research was partially funded by the EU ECSEL project DAIS under grant agreement No. 101007273. The opinions expressed here are solely those of the authors and do not necessarily reflect the views of the European Commission, which is not responsible for any use of the information provided.

REFERENCES

[1] R. A. Khalil, E. Jones, M. I. Babar, T. Jan, M. H. Zafar, and T. Alhussain, “Speech emotion recognition using deep learning techniques: A review,” IEEE Access, vol. 7, pp. 117 327–117 345, 2019.
[2] M. B. Akçay and K. O˘guz, “Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers,” Speech Communication, vol. 116, pp. 56–76, 2020.
[3] P. Chhikara, P. Singh, R. Tekchandani, N. Kumar, and M. Guizani, “Federated learning meets human emotions: A decentralized framework for human–computer interaction for iot applications,” IEEE Internet of Things Journal, vol. 8, no. 8, pp. 6949–6962, 2020.
[4] J. L. Kröger, O. H.-M. Lutz, and P. Raschke, “Privacy implications of voice and speech analysis–information disclosure by inference,” Privacy and Identity Management. Data for Better Living: AI and Privacy: 14th IFIP WG 9.2,

9.6/11.7, 11.6/SIG 9.2. 2 International Summer School, Windisch, Switzerland, August 19–23, 2019, Revised Selected Papers 14, pp. 242–258, 2020.

[5] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," A Practical Guide, 1st Ed., Cham: Springer International Publishing, vol. 10, no. 3152676, pp. 10–5555, 2017.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Artificial intelligence and statistics. PMLR, 2017, pp. 1273– 1282.

[7] S. Latif, S. Khalifa, R. Rana, and R. Jurdak, "Federated learning for speech emotion recognition applications," in 2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). IEEE, 2020, pp. 341–342.

[8] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in 2019 IEEE symposium on security and privacy (SP). IEEE, 2019, pp. 739–753.

[9] M. S. Jere, T. Farnan, and F. Koushanfar, "A taxonomy of attacks on federated learning," IEEE Security & Privacy, vol. 19, no. 2, pp. 20–28, 2020.

[10] Z. Xiong, Z. Cai, D. Takabi, and W. Li, "Privacy threat and defense for federated learning with non-iid data in aiot," IEEE Transactions on Industrial Informatics, vol. 18, no. 2, pp. 1310–1321, 2021.

[11] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K.-Y. Lam, "Local differential privacy-based federated learning for internet of things," IEEE Internet of Things Journal, vol. 8, no. 11, pp. 8836–8853, 2020.

[12] K.Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3454–3469, 2020.

[13] M. Kim, O. Günlü, and R. F. Schaefer, "Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication," in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 2650–2654.

[14] M. A. Pathak, Privacy-preserving machine learning for speech processing. Springer Science & Business Media, 2012.

[15] T. Feng, R. Peri, and S. Narayanan, "User-level differential privacy against attribute inference attack of speech emotion recognition in federated learning," arXiv preprint arXiv:2204.02500, 2022.

[16] A. A. Alnuaim, M. Zakariah, A. Alhadlaq, C. Shashidhar, W. A. Hatamleh, H. Tarazi, P. K. Shukla, and R. Ratna, "Human-computer interaction with detection of speaker emotions using convolution neural networks," Computational Intelligence and Neuroscience, vol. 2022, 2022.

[17] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, 2015, pp. 1322–1333.

[18] V. Tsouvalas, T. Ozcelebi, and N. Meratnia, "Privacy-preserving speech emotion recognition through semi-supervised federated learning," in 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops). IEEE, 2022, pp. 359–364.

[19] Y. Chang, S. Laridi, Z. Ren, G. Palmer, B.W. Schuller, and M. Fisichella, "Robust federated learning against adversarial attacks for speech emotion recognition," arXiv preprint arXiv:2203.04696, 2022.

[20] T. Tuncer, S. Dogan, and U. R. Acharya, "Automated accurate speech emotion recognition system using twine shuffle pattern and iterative neighborhood component analysis techniques," Knowledge-Based Systems, vol. 211, p. 106547, 2021.

[21] P. Liu, X. Xu, and W. Wang, "Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives," Cybersecurity, vol. 5, no. 1, pp. 1–19, 2022.

[22] R. Bassily, "Linear queries estimation with local differential privacy," in The 22nd International Conference on Artificial Intelligence and Statistics. PMLR, 2019, pp. 721–729.

[23] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, 2016, pp. 308–318.

[24] A. Balador, S. Sinaei, M. Pettersson, and I. Kaya, "Dais project - distributed artificial intelligence systems: Objectives and challenges," in 26th Ada-Europe International Conference on Reliable Software Technologies (AEiC'22), 2022.

[25] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, "Crema-d: Crowd-sourced emotional multimodal actors dataset," IEEE transactions on affective computing, vol. 5, no. 4, pp. 377–390, 2014.

[26] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor," in Proceedings of the 18th ACM international conference on Multimedia, 2010, pp. 1459–1462.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

Scan to save the contact details

www.ijircce.com